

CLAIMS

1 1. A method in a computer system for determining resolution of attributes
2 of a program, the method comprising:
3 providing a program having interactions, each interaction having commands
4 with attributes;
5 identifying a sequence of interactions of the program; and
6 for each interaction in the identified sequence,
7 for each command of the interaction,
8 for each input attribute of the command,
9 identifying an output attribute corresponding to the input attribute; and
10 setting the resolution of the input attribute to the resolution of the
11 identified output attribute; and
12 for each output attribute of the command, setting the resolution of the
13 output attribute to resolved.

1 2. The method of claim 1 including reporting input attributes whose
2 resolution is set to unresolved.

1 3. The method of claim 2 including suppressing the reporting of input
2 attributes that may be resolved by user input.

1 4. The method of claim 2 including suppressing the reporting of input
2 attributes of primitive types.

1 5. A method for verifying resolution of input parameters of functions of a
2 computer program before executing the computer program, the method comprising:
3 providing a path of execution of the computer program, the path of execution
4 identifying a sequence of functions of the computer program; and

5 for each function identified in the provided path of execution, processing the
6 function by
7 for each input parameter of the function, indicating that the input
8 parameter is resolved when a corresponding output parameter has been indicated as resolved
9 when a function in the path of execution was previously processed; and
10 for each output parameter of the function, indicating that the output
11 parameter is resolved.

1 6. The method of claim 5 wherein the computer program is a command-
2 based application wherein the functions are methods of objects corresponding to the
3 commands.

1 7. The method of claim 6 wherein the commands are organized into
2 interactions.

1 8. The method of claim 6 wherein the parameters are attributes of the
2 objects.

1 9. The method of claim 8 wherein the objects have set and get methods for
2 setting and getting attribute values.

1 10. The method of claim 8 wherein the attribute values are set with an
2 assignment statement.

1 11. The method of claim 6 wherein each object has a perform method for
2 performing a behavior associated with the command.

1 12. The method of claim 5 wherein input and output parameters correspond
2 when they have the same name.

1 13. The method of claim 5 including creating a list of each parameter of
2 each function processed, the list indicating resolution of the parameter, and including
3 outputting an indication of resolution of each parameter based on the created list.

1 14. The method of claim 5 wherein the computer program is specified by an
2 interaction-based definition, wherein interactions include commands, and wherein each
3 command has a corresponding object with attributes.

1 15. The method of claim 14 wherein each command is defined by a
2 descriptor that optionally provides aliasing for names of attributes.

1 16. The method of claim 14 wherein each command is defined by a
2 descriptor that optionally provides a constant value for an attribute.

1 17. A computer system for verifying resolution of input parameters of
2 functions of a computer program before executing the computer program, comprising:
3 means for selecting each function in execution order; and
4 means for processing each selected function by for each input parameter of the
5 function, indicating that the input parameter is resolved when a corresponding output
6 parameter has been indicated as resolved when a function of the computer program was
7 previously processed and for each output parameter of the function, indicating that the output
8 parameter is resolved.

1 18. The computer system of claim 17 wherein the computer program is a
2 command-based application wherein the functions are methods of objects corresponding to
3 the commands.

1 19. The computer system of claim 18 wherein the commands are organized
2 into interactions.

1 20. The computer system of claim 18 wherein the parameters are attributes
2 of the objects.

1 21. The computer system of claim 20 wherein the objects have set and get
2 methods for setting and getting attribute values.

1 22. The computer system of claim 18 wherein each object has a perform
2 method for performing a behavior associated with the command.

1 23. The computer system of claim 17 wherein input and output parameters
2 correspond when they have the same name.

1 24. The computer system of claim 17 including means for creating a list of
2 each parameter of each function processed, the list indicating resolution of the parameter,
3 and including means for outputting an indication of resolution of each parameter based on
4 the created list.

1 25. The computer system of claim 17 wherein the computer program is
2 specified by an interaction-based definition, wherein interactions include commands and
3 wherein each command has a corresponding object with attributes.

1 26. The computer system of claim 25 wherein each command is defined by
2 a descriptor that optionally provides aliasing for names of attributes.

1 27. The computer system of claim 25 wherein each command is defined by
2 a descriptor that optionally provides a constant value for an attribute.

1 28. A computer system for processing each function of a computer program
2 prior to runtime by for each input parameter of the function, determining whether a source of
3 the input parameter would be resolved during execution of the computer program and for

4 each output parameter of the function, indicating that the output parameter is resolved
5 wherein output parameters are sources of input parameters.

1 29. The computer system of claim 28 wherein the computer program is a
2 command-based application wherein the functions are methods associated with objects
3 corresponding to the commands.

1 30. The computer system of claim 29 wherein the commands are organized
2 into interactions.

1 31. The computer system of claim 29 the parameters are attributes of the
2 objects.

1 32. The computer system of claim 31 wherein the objects have set and get
2 methods for setting and getting attribute values.

1 33. The computer system of claim 29 wherein each object has a perform
2 method for performing a behavior associated with the command.

1 34. The computer system of claim 28 wherein the computer program is
2 specified by an interaction-based definition, wherein interactions include commands and
3 wherein each command has a corresponding object with attributes.

1 35. The computer system of claim 34 wherein each command is defined by
2 a descriptor that optionally provides aliasing for names of attributes.

1 36. The computer system of claim 34 wherein each command is defined by
2 a descriptor that optionally provides a constant value for an attribute.

1 37. A computer-readable medium containing instructions for controlling a
2 computer system to determine prior to runtime resolution of parameters of functions of a
3 computer program, by a method comprising:

4 identifying a path of execution of the computer program, the path of execution
5 having of functions of the computer program;
6 for functions in the identified path of execution,
7 indicating that an input parameter of the function is resolved when a
8 corresponding output parameter has been indicated as resolved; and
9 indicating that an output parameter is resolved.

1 38. The computer-readable medium of claim 37 including indicating that an
2 input parameter of a primitive type may be resolved by a user at runtime.

1 39. The computer-readable medium of claim 37 wherein the computer
2 program is a command-based application where the functions are methods associated with
3 objects corresponding to the commands.

1 40. The computer-readable medium of claim 39 wherein the commands are
2 organized into interactions.

1 41. The computer-readable medium of claim 39 wherein the parameters are
2 attributes of the objects.

1 42. The computer-readable medium of claim 41 wherein the objects have set
2 and get methods for setting and getting attribute values.

1 43. The computer-readable medium of claim 37 wherein the computer
2 program is specified by an interaction-based definition, wherein interactions include
3 commands and wherein each command has a corresponding object with attributes.

1 44. The computer-readable medium of claim 43 wherein each command is
2 defined by a descriptor that optionally provides aliasing for names of attributes.

1 45. The computer-readable medium of claim 43 wherein each command is
2 defined by a descriptor that optionally provides a constant value for an attribute.